

# 1. Introducción

Git es un programa para control de versiones. Nos permite crear un repositorio local (análogo a nuestra carpeta en Dropbox) en donde podremos realizar todos los cambios locales que queramos para, posteriormente subirlo a un repositorio global.

La ventaja que presenta este sistema es que lleva un control de cada una de las versiones que se van añadiendo, de modo que es muy sencillo volver a una versión anterior (aún si se borran todos los archivos por error). Además, si dos usuarios realizan cambios simultáneos y lo suben al repositorio global, el programa creará dos ramas (branches) de modo que pueden volver a unirse en una única “rama” de versiones.

He creado el repositorio global en `rtnucm:/homenfs/rg/global_repository.git` en este repositorio **no se puede copiar ningún archivo de manera usual**. Este repositorio sirve como repositorio global al que accederemos desde nuestros repositorios locales (para hacer pruebas yo tengo uno en quercus y otro en mi carpeta personal de rtnucm).

## 1.1. Crear un repositorio local

Para crear un repositorio local deben seguirse los siguientes pasos:

- Crear una carpeta para el repositorio.
- Crear el repositorio con

```
1  git init
```

- Establecer un usuario y un correo (opcional) para identificar quién ha realizado cada cambio en la versión.

```
1  git config --global user.name "<username>"
2  git config --global user.mail <usermail>@gmail.com [OPCIONAL]
```

## 1.2. Añadir un repositorio global

Para añadir un repositorio global (en este caso pondré de ejemplo el que he creado en rtnucm) se seguirán los siguientes pasos:

- Añadimos el repositorio global con un identificador propio <repository name>

```
1  git remote add <repository name>
    ssh://<user_name>@rtnucm.fis.ucm.es:/homenfs/rg/global_repository.git
```

- Importamos todo lo que tenga el repositorio para evitar problemas de branching iniciales

```
1 git fetch <repository name>
2 git pull <repository name> <branch>
```

Por defecto, <branch>=master, es decir, la rama oficial de las versiones es master, solamente hay que modificar el nombre de esa rama cuando queramos tener varias versiones paralelas.

### 1.3. Añadir/borrar/mover archivos

Para añadir archivos a nuestro repositorio hay que realizar los siguientes pasos:

- Importamos del repositorio global los archivos existentes (esta medida es únicamente de seguridad para evitar que se produzcan ramificaciones, ya que si alguien ha realizado cambios pero no los importamos habrá conflictos en la rama principal master).

```
1 git fetch <repository name>
2 git pull <repository name> <branch>
```

- Copiamos al directorio en el que se encuentra nuestro repositorio local todos los archivos que queramos añadir.
- Añadimos los cambios que queremos actualizar

```
1 git add <file names>
2 git rm <file names>
3 git mv <file name 1> <file name 2>
```

- Realizamos los cambios de forma local

```
1 git commit -a -m "Descripcion del cambio realizado"
```

La opción m"nos permite incluir un mensaje para que los otros usuarios sepan qué cambios se han llevado a cabo en esa versión.

- Mandamos los cambios al repositorio global.

```
1 git push <repository name> master
```

### 1.4. Borrar repositorio

Para borrar un repositorio remoto únicamente tenemos que escribir en nuestro repositorio local

```
1 git remote rm <repository name>
```

Esto únicamente borra el acceso de nuestro repositorio local al repositorio global, en ningún caso borra los archivos del repositorio.

## 1.5. Algunas órdenes útiles

Existen algunos comandos útiles a la hora de trabajar con git:

- Esta orden nos muestra por pantalla los cambios realizados (con el mensaje correspondiente), el autor del cambio y la fecha del mismo.

```
1 git log
```

- Este comando te permite saber en qué estado están los archivos de tu repositorio (si se han llevado los cambios al repositorio global, si está actualizado, etc..)

```
1 git status
```

- Si realizas cambios con la orden commit y te das cuenta de que te falta algún archivo, puedes incluir nuevos archivos sin necesidad de hacer todo el proceso de update de nuevo con:

```
1 git commit -m 'initial commit'
2 git add <forgotten file>
3 git commit --amend
```

- Este comando te da una lista de los repositorios remotos que tienes asociado a tu repositorio local

```
1 git remote
```